

PROJECT ABSTRACT

Clipboard History Manager Desktop App

A desktop clipboard history manager with full-text search, pinning, AES-256 encrypted storage and optional cross-device sync. Delivered built-to-order with the full application, installers, documentation, report, PPT and viva kit.

01. Title

Clipboard History Manager Desktop App — a system-tray desktop application that records everything copied on the computer and makes the full history searchable, pinnable and restorable with a single hotkey.

02. Introduction / Background

Operating systems keep only the most recent clipboard item, so anything copied earlier is lost the moment something new is copied. Developers, writers and students re-type terminal commands, code blocks and reference links daily because of this limitation. A clipboard history manager listens to OS clipboard events and preserves every entry — text, code, links and images — with its type, source application and timestamp, in a searchable, encrypted local database.

03. Problem Statement

There is no built-in, private way to recover something copied an hour ago. Existing tools are either cloud-first (sending copied data to third-party servers) or lack search, pinning and encryption. This project builds a local-first desktop application that captures clipboard history, indexes it for instant search, protects it with AES-256 encryption, and optionally syncs it across the user's devices over an end-to-end encrypted channel.

04. Objectives

- Capture clipboard changes system-wide (text, code, links, images) with source-app and timestamp metadata.
- Provide instant full-text search over the history using SQLite FTS5 indexing.
- Offer pinning, favorites and per-type filters with a global hotkey (Ctrl+Shift+V) paste overlay.
- Encrypt the local store with AES-256 (Argon2id key derivation) and sync across devices end-to-end encrypted.
- Ship installers for Windows, macOS and Linux with a user manual and complete viva documentation.

05. Proposed Methodology

The application is structured as a background monitor service plus a Qt desktop UI. OS-specific clipboard listeners (Win32 viewer chain, NSPasteboard notifications, X11

selections) feed a capture pipeline that classifies each entry and writes it to an encrypted SQLite database. An FTS5 index is updated incrementally for search. The hotkey overlay is summoned globally; double-clicking an entry pastes it via simulated keystrokes. An optional sync relay exchanges only ciphertext between a user's devices, merging by timestamp.

06. System Architecture / Working

Monitor layer (OS clipboard hooks) → capture pipeline (type classification, metadata) → encrypted SQLite store + FTS5 index → Qt UI (history window, hotkey overlay, settings) → optional sync relay (E2E encrypted). Retention sweeps delete unpinned entries older than the configured window; pinned and favorite items are exempt.

07. Hardware / Software Requirements

- A Windows 10/11, macOS 13+ or Ubuntu 22.04+ machine for development and use.
- Python 3, PyQt6, SQLite with FTS5, cryptography and Argon2id libraries, pynput, PyInstaller.
- No special hardware; expected resident footprint is approximately 60–90 MB.

08. Expected Outcomes

A working desktop application that records clipboard history, searches it instantly, pastes via global hotkey, encrypts storage, and optionally syncs across devices — plus installers, a user manual, a project report, PPT and viva Q&A.

09. Applications

- Developers reusing terminal commands, code snippets and API keys.
- Writers and students recovering copied quotes, links and references.
- Office staff reusing formatted text and boilerplate across documents.

10. Future Scope

- Cloud backup of the encrypted database with user-held keys.
- Browser-extension capture for per-tab attribution.
- Smart suggestions that surface frequently pasted snippets contextually.